

Advantage

INTRODUCTION

The *Advantage* is a spreadsheet with integrated graphics and database management facilities. It provides many of the features normally found only in spreadsheet programs costing substantially more. These features include:

- Spreadsheet size of up to 32000 rows by 32000 columns
- Sparse memory matrix - empty cells consume no memory
- Minimal recalculation - only cells which have been affected by other changes on the spreadsheet are calculated, resulting in very short recalculation times
- ARexx interface with a special macro command language
- Built-in macro facility allows sequences of keystrokes and menu commands to be recorded and played back
- Wide selection of chart types and chart formatting options
- Multiple (named) databases and selection criteria on a single spreadsheet
- Named cells and ranges
- User defined functions
- Large selection of built-in functions (over 100) including statistical, trigonometric and financial functions
- Runs on Workbench or custom screen with up to 16 colors
- Supports overscan displays through use of the MoreRows utility

Overview of this Manual

No doubt you are eager to begin using the *Advantage*. We want you to get the most out of this program with the least possible delay. To that end, here is a brief overview of this section of the manual and our suggestions on how to make the best use of it.

Introduction

Where you are right now! Whether you are an experienced spreadsheet user or a novice, you should read the remainder of this chapter first to familiarize yourself with the way in which the *Advantage* lays out its spreadsheet windows.

Tutorial

An in-depth tutorial aimed at familiarizing the novice with spreadsheet operations. If you have used a spreadsheet before, you can skip this chapter. You may wish to scan it briefly, however, since it contains a few "tips" specific to the *Advantage*.

Charts

Here you will find an comprehensive look at the manner in which the *Advantage* can represent spreadsheet information in graphic form. The variety of charts, and the options provided for each are reviewed in this chapter.

Database Operations

Learn how sections of a spreadsheet can act as a database in this chapter. Creating databases, as well as operations affecting databases, such as sorting and indexing are covered here.

Macros

In this chapter you will find a full description of the macro facilities of the *Advantage*. This time-saving feature allows a complex series of mouse selections and typed characters to be recalled with a single key stroke.

ARexx Interface

This chapter will be of great interest if you own a copy of the ARexx language. It describes the macro language and ARexx interface provided by the *Advantage*. The unique *Script* button features of the *Advantage* are also detailed in this chapter. Imaginative use of ARexx and *Script*

buttons can turn the *Advantage* in a powerful interactive presentation tool.

Menu Reference

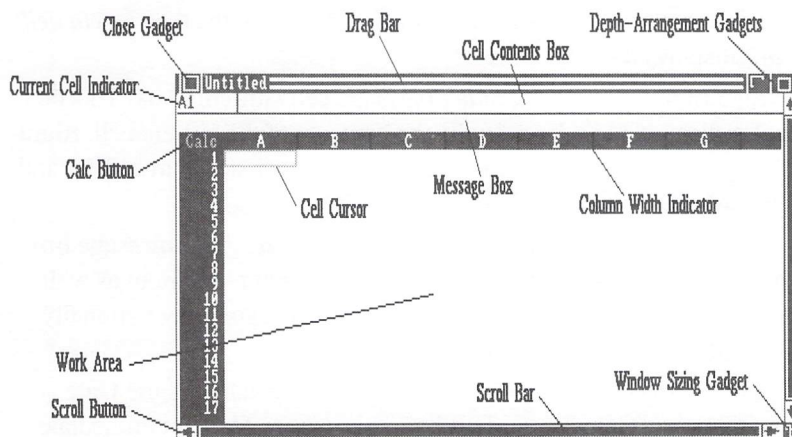
In this chapter, all the menus of the *Advantage* are listed and explained sequentially. You might find this useful, once you become comfortable with the program, to serve as a quick reminder of the details of a specific function or feature.

Scattered throughout the manual are various tips and techniques that you might not discover in using the *Advantage* on your own, even if you are an experienced user. You can scan the manual and read through these quickly by looking for this marker in the margin: **TIP**.

A number of example spreadsheets are provided in the Examples drawer on your distribution disk. These spreadsheets augment the manual by providing working examples of the various functions available in the *Advantage*. We encourage you to look at these examples as a means of clarifying some of the more difficult concepts covered in the manual.

The Spreadsheet Window

To begin, follow the directions outlined in "Getting Started" to get the *Advantage* running on your Amiga. When the *Advantage* screen appears a blank spreadsheet window will automatically be opened on the screen. The window contains the standard window close, drag bar, depth-arrangement and window sizing gadgets used in many Amiga



application programs. (If you are not familiar with the use of these gadgets, refer to the chapter "Using the Workbench" in the "Introduction to Amiga" manual that came with your computer.) The spreadsheet name appears at the left end of the window drag bar. New spreadsheets, such as this one, are labelled "Untitled" until they are saved under a name of your choosing. The majority of the window is taken up by the spreadsheet **work area**. The work area is divided up into a grid of boxes referred to as **cells**. Each column of the grid is labelled with an identifying letter or letters (along the top of the work area) and each row is labelled with a number (at the left end). The individual cells are situated at the intersection of these columns and rows and each cell can be uniquely identified by the appropriate column and row labels. For example, the cell in the upper left corner of our "Untitled" spreadsheet lies at the intersection of column "A" and row "1" and is, therefore, referred to as cell "A1".

The column labels, along the top of the work area, are separated by vertical black bars called **column width markers**. These markers indicate the current width of each column and may also be used to adjust the width, within allowable limits.

At the top left of the window, just below the close gadget, is the **current cell indicator**. The name of the currently selected cell is displayed here. The current cell is outlined by a red box in the spreadsheet work area. This box is similar to the cursor used to mark your current position in a text editor. For the sake of clarity, we'll refer to this box as the **cell cursor**. This will differentiate it from the regular text cursor which is also used by the *Advantage*. You will note that the *cell cursor* is currently located in the upper left corner of the work area, at cell A1. Accordingly, the *current cell indicator* displays the name "A1".

To the right of the *current cell indicator* is the **cell contents box**. This box is used for displaying and modifying the contents of the current cell. Right now the entire spreadsheet is empty. There is nothing stored in cell A1 and the **cell contents box** is blank.

Below the *current cell indicator* and *cell contents box* is the **message box**. This box is used to display error messages and prompts for input as well as user defined messages (called "cell notes") which you may optionally attach to specific cells in the spreadsheet.

At the intersection of the work area column and row labels is the **Calc button** (short for "Calculate"). This button is used to force an immediate recalculation of all formulas on the spreadsheet which currently require

updating. As we mentioned before, the *Advantage* will only recalculate cells which may have been affected by other changes on the spreadsheet. This procedure can dramatically decrease the time it takes to recalculate a spreadsheet.

Along the bottom and right borders of the spreadsheet window are located the work area scroll gadgets. Each set of scroll gadgets consists of two **scroll buttons** surrounding a **scroll bar**. These gadgets allow you to change the selection of the current cell and to view different portions of the work area. In addition, the scroll bars provide an indication of where you are within the work area as well as how much of the work area you can view at one time.

TUTORIAL

In this chapter we will take a detailed look at creating a simple spreadsheet application from the ground up. If you've never used a spreadsheet before, this walk-through will quickly acquaint you with the basics of spreadsheet operation as well as some of the unique features of the *Advantage*. Before starting this tutorial you should be familiar with the terminology explained in the "Introduction" which describes the layout of the *Advantage's* spreadsheet windows.

A spreadsheet is a special kind of application program. Rather than providing a complete "prepackaged" solution to your problem, it provides only the "framework". It's up to you to fill in the blanks and, in effect, to "program" the spreadsheet for your special needs. But you don't need years of training to be able to program a spreadsheet. You only need to know what calculations you would normally make by hand and then simply transfer the appropriate equations to the spreadsheet in an organized fashion. Let's create a simple spreadsheet to find out how this is done.

If you have not already done so, start the *Advantage* running on your Amiga as described in "Getting Started".

The ABC Letter Company

Let's say you are the president of the ABC Letter Company which manufactures letters of the alphabet. With people continually wanting to write books, magazines and newspapers, there is a steady demand for your product. Business is brisk and you decide to begin tracking your company profits on your computer. Each month, you know how much money you took in from sales (income) and how much money was spent on manufacturing and shipping of your letters (expenses). You also know that your net profit is just the difference between these two values, income minus expenses.

To begin setting up your spreadsheet, put a title at the top. We want to center the title over the other information we're going to display so we'll put it in cell D1. You can move the *cell cursor* to cell D1 in various ways. The easiest way is to simply click on cell D1 with the mouse. The *cell cursor* immediately jumps to the selected cell and the *current cell indicator* displays the name of the cell. You can also use the cursor keys on the keyboard to move the *cell cursor*, one step at a time, in the indicated direction. If you start at cell A1 and press the right arrow key three times, you'll wind up at D1. Lastly, the *Scroll* buttons around the border of the work area operate in the same manner as the cursor keys. If you start at cell 1, you can click on the "*right*" *Scroll* button (located near the bottom right corner of the window) three times, you'll again wind up at D1.

However you decide to do it, position the *cell cursor* over cell D1. Make sure you have selected the correct cell by checking the name in the *current cell indicator*. Now type in the title for this spreadsheet - "ABC Letter Company" (the quotes are not required). When you type the first letter ("A"), you will note that a cursor automatically appears in the *cell contents box* and the text you enter is displayed there. You can edit the text in the *cell contents box* using the CTRL-left and CTRL-right cursor key combinations, the backspace and delete keys. When you have entered the complete title, press RETURN to copy it to the spreadsheet.

The title will appear in cell D1 and the *cell cursor* will automatically advance to the "next" cell in sequence. In this case, the next cell is one row down, at D2. A *Preference* option is provided by the *Advantage* that allows you to modify the cell entry sequence (the direction and

amount by which the *cell cursor* will move each time you enter data into a cell). You may sometimes find it convenient to change the entry sequence but for now, leave it as is.

You may have noticed that the title we entered is wider than column D. The **Advantage** has displayed the entire title by allowing it to overflow into column E. When a label (text) is placed on the worksheet it is allowed to overflow to the right, if necessary, until an "occupied" cell is encountered. If we were to enter something into cell E1, the label in D1 would then be truncated at the right edge of column D. This "overflow" capability applies only to label cells. As we will see, there are other types of cells (formula and value cells, for example) that do not behave this way. For these other cell types, the cell contents are always confined to the limits of the column. If the required data cannot be displayed within those limits, a series of asterisks (*) will be shown instead, to let you know that there is an overflow problem.

Now let's label the columns where the monthly income and expense figures will be entered into the spreadsheet. Move the *cell cursor* to B3. Enter "Jan" into this cell. Continue by entering the three letter month names into cells C3, D3, etc. Stop once you have entered "Jun" into G3 - we'll only set up the spreadsheet for half a year for this example. Put the label "Total" into H3. This column will display the total income, expenses and profit for the half year. Since column H is currently off the screen, you must use the keyboard *cursor keys* or the *Scroll* buttons to position the *cell cursor* on this cell. When the *cell cursor* reaches the right edge of the work area and you continue moving it to the right, the work area will scroll to the left to reveal the offscreen portion of the spreadsheet.

Label the rows of the spreadsheet as follows: "Income" in cell A4, "Expenses" in A5 and "Net Profit" in A7 (Note: A6 is left blank!).

Now start entering the actual income and expense figures into the spreadsheet. Enter the following table of values:

	Jan	Feb	Mar	Apr	May	Jun
Income	800	1000	800	1400	2000	1500
Expenses	700	600	1000	1200	1500	1250

These cells which you have just entered are referred to as "value" cells because they contain constant numeric values.

Formula Cells

Now let's create our first "formula" cell. Move the cell cursor to B7. We want this cell to display the company's net profit for the month of January. We know the net profit is calculated by:

Net Profit = Income - Expenses

The income for January is in cell B4 and the expenses are in B5. By substituting these cell names into the formula we get:

Net Profit for January = B4 - B5

The *Advantage* doesn't need to know what it is calculating (in this case, the net profit for January). To tell the *Advantage* to calculate a value in a cell, enter the equals sign and the formula to the right of the equals sign. Enter "=B4-B5" into cell B7. When you press RETURN the calculation is immediately performed and the result is displayed - "100" in this case. If you now reselect cell B7 you'll see that, although the result of the formula is displayed on the spreadsheet, the formula assigned to the cell is displayed in the *cell contents box*.

Formula cells are the "heart" of a spreadsheet. Because the formula refers to cells (as opposed to constant values), any changes to the contents of those cells will be reflected in the formula cell when the spreadsheet is recalculated. For example, try changing the value in B4 to 900. When you press RETURN, the number displayed in B7 will change to 200. The value of this feature becomes more apparent in applications where many calculated values depend on the contents of a single cell.

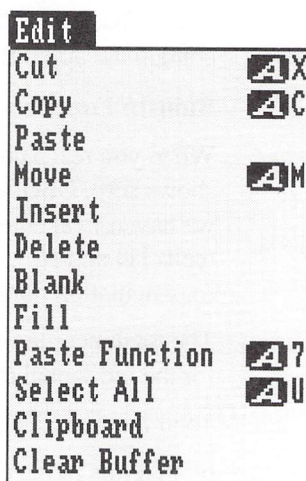
Spreadsheet Editing

To separate the income and expense figures from the calculated profit figures, put a row of equals signs at the bottom of each column. Move to cell B6 and enter " =====" (a space and 8 equals signs). Note that you must precede the first equals sign with a space. Otherwise, the *Advantage* will interpret the entry as a formula (an incorrect formula!) rather than a label. Instead of repeating this procedure for each column, let's use the *Advantage's* editing facilities to make copies of this label.

Select cell B6 and then select *Edit/Copy* from the menu. This command copies the selected cell to a temporary work space.

Now select cell C6 and then *Edit/Paste/Full* from the menu. The contents of the work space is copied back into the selected cell. Use *Edit/Paste/Full* to make copies in cells D6 through H6.

The same formula that we used to calculate the net profit for January can be applied to the remaining months as well as to the half year totals. However, the formula must be adjusted to refer to the appropriate income and expense cells for that month. Fortunately, the formula can be easily copied and adjusted at the same time by the *Advantage's Fill* command.



Use the mouse to select cell B7. Instead of releasing the mouse button, hold it down and drag the mouse to the right. As you drag, the cell cursor will expand to encompass the cells to the right of B7. Continue dragging the mouse until the **cell cursor** surrounds the entire row of cells between B7 and H7, then release the mouse button. This procedure is called **extended selection** and it allows you to select a "range" of cells rather than a single cell. (We'll talk more about ranges shortly.) You can also perform an extended selection by clicking on the cell at one corner of the desired range, moving the mouse to the diagonally opposite corner of the range and then pressing the SHIFT key while you click on that cell. When a range of cells is selected, the current cell is the first cell selected in the range. In this case, since B7 was selected first, you will note that the *current cell indicator* shows "B7".

Select *Edit/Fill/Right* from the menu. The formula in B7 is copied into successive cells to the right until the end of the selected range is reached. As the formula is copied, all cell references are updated to offset them by the same amount by which the formula has been moved. For example, when the formula " $=B4-B5$ " is moved one column to the right, to cell C7, the cell references are also "moved" over by one column. Select cell C7 and you will see that it now contains the formula

"=C4-C5". Check the contents of cells D7 through H7 to verify that they contain the appropriately modified formula as well.

Built-In Functions

When you reach column H, you will notice that the total profit figure shows zero. Obviously, this is not the correct answer. The problem is that we haven't yet entered the formulas for the total income and expenses (cells H4 and H5). Since these cells are blank, they have a value of zero for calculation purposes and, therefore, the total profit is zero.

The total income for the half year is simply the sum of the income figures for the individual months. This could be written:

Total Income = "Jan" Income + "Feb" Income + + "Jun" Income

or, in terms of the cell names:

Total Income = B4 + C4 + D4 + E4 + F4 + G4

Although there is nothing wrong with entering the formula as shown above, the *Advantage* provides a better way, in the form of the "SUM" **built-in function**. The SUM function returns the total value of a list of numbers, cell references and calculations supplied as arguments. For example, the total income could be calculated as:

Total Income = SUM(B4, C4, D4, E4, F4, G4)

To use a built-in function in one of our formulas, we first enter the function name ("SUM"), followed by an opening bracket ("("), followed by the appropriate function arguments and then finish with a closing bracket (")"). For a complete list of built-in functions and their required arguments, refer to the *Functions* section. In this case, the arguments for SUM consist of a list of cell names. It is also possible to include constant values, formulas and function references in the list. For example, we might write this formula:

Some Value = 15.6 + (2 x One Value) + (Multiplier x Other Value)

like this:

Some Value = SUM(15.6, 2*D7, C4*G9)

Since it is also acceptable to nest one function reference inside another, the above formula could also be written as:

Some Value = SUM(15.6, SUM(2*D7, C4*G9))

Both formulas will produce the same answer but the first one, being simpler, is the form you're more likely to use.

Cell Ranges

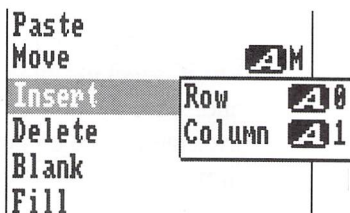
Going back to our example using SUM for the Total Income, you can see that this form really isn't any simpler than the form which uses straight addition. We can make another change that will make the use of SUM more efficient. Since the values to be added are all located together in the work area, we can replace the list of individual cells names with a **cell range**. A cell range is used to delimit a rectangular region within the work area and includes all cells within that region. A range consists of the names of two cells at diagonally opposite corners of the desired region, separated by a colon. For example, the range "A1:C3" is a shorthand method of specifying all of the cells: A1, A2, A3, B1, B2, B3, C1, C2, C3.

By using a range in our Total Income formula, we can reduce it to:

Total Income = SUM(B4:G4)

Besides providing a convenient shorthand method of writing a list of cell names, cell ranges have another advantage that is not obvious to first time spreadsheet users. This feature is best illustrated by example. Let's say you want to add up a column of numbers in cells A9 through A11. Enter some values in these cells (the actual values don't matter for this example). Put a row of equal signs into cell A12 (to separate the values from the answer) and enter the formula "=SUM(A9:A12)" into cell A13. Note that since the label in cell A12 has a value of zero, it does not affect the calculation of the total. However, it's useful to include this cell in the range for reasons that will become apparent.

Now, let's say that at some later date you decide to add another value to the column of numbers. You can create an opening for this new value by selecting cell A12 and then using *Edit/Insert/Row* from the menu. This command inserts a new blank row at row 12 and pushes rows greater than this one, down one row. At this point, cell A13 contains the row of equals signs and A14 contains the column total. Now enter your new value in A12. The total will update to reflect the addition of the new value. Select cell A14 and you will see that the formula has been



modified to account for the insertion of a new row. It now reads
"=SUM(A9:A13)".

This is an important feature of ranges used as function arguments. **Range arguments are updated whenever rows or columns within the boundaries of the range are inserted or deleted from the spreadsheet.** If you had used the more direct formula
("=A9+A10+A11+A12") to calculate the total, you would have had to update the formula yourself. Now you can see why cell A12 (equals signs) was included in the range. This allowed us to add a new cell to the bottom of the column. If the range had stopped at A11, inserting a row at A12 would not have updated the formula. With a little forethought, you can save yourself a lot of work!

Now let's finish off our ABC Letter Company example. Select cell H4 and type in "=SUM(", but don't press RETURN. Now, rather than typing in the cell range, let's make use of one of the *Advantage's* editing shortcuts. With the mouse, click on cell B4. Hold the left mouse button down and drag the mouse to cell G4. When you release the left mouse button, the range you have selected ("B4:G4") will be written into the *cell contents box* at the current position of the *text cursor*. Complete the formula by entering the closing bracket (")") and pressing RETURN.

The total expenses are calculated in the same manner as the total income. Into cell H5, enter "=SUM(B5:G5)". Now click the *Calc* button to calculate the net profit for the first half of the year.

Keeping Up Appearances

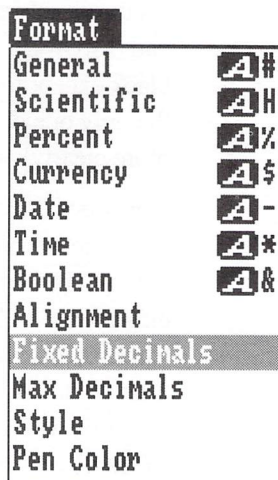
TIP

The mechanics of our spreadsheet are complete - all of the labels, values and formulas have been entered. However, the *Advantage* provides a few commands that will let us not only improve the appearance of the spreadsheet, but also indicates important information. Let's take a look at a few of these features.

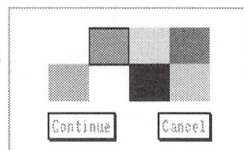
You may have noticed that the final "t" is missing from the "Net Profit" label in cell A7. Column A is too narrow to display this label and, since the next cell to the right is occupied, the label has been truncated to the column width. A quick adjustment can remedy this problem.

Select the *column width marker* between columns A and B. Hold the left mouse button down to keep the *marker* selected. A red line appears, running down the spreadsheet below the *marker*, and the number "9" is displayed in the *message box*. While keeping the mouse button depressed, drag the *marker* to the right. As you do this, the number in the *message box* will increase. This number indicates the new column width, if you were to release the *marker* at that point. The column can be stretched as wide as 32 characters and shrunk to as little as 1 character. Adjust the column width to 10 characters and then release the *marker*. Now the "Net Profit" label is completely visible.

Use extended selection to select the range of cells displaying numeric values on the spreadsheet (cells B4:H7). Select *Format/Fixed Decimals* and then *Format/Max Decimals/2* from the menu. Now all the numbers are displayed to two decimal places - a more common format for dollar amounts. You might also want to try *Format/Currency*. This will again display the values to two decimal places, but a leading dollar sign is added to each number. In this case the dollar signs tend to clutter the display so it's better to leave them out. Notice that the labels in row 6, although included in the current range, are not affected by the selections we made from the *Format* menu. These particular format items only affect cells displaying numbers.



Select the label cells at the top of the columns of monthly figures (cell range B3:H3). First, select *Format/Alignment/Centered* to center the labels above their respective columns. Now select *Format/Pen Color* from the menu. The **Color Selection Requester** will appear. Click on the blue square and then the **Continue button**. All of the selected labels will change to the blue color. Now select *Format/Style/Italic*. The labels change to italic text. You can choose any pen color and text style to help differentiate the various items on your spreadsheet and make them stand out. Try to use a little restraint, however, when applying these options or you could end up with a very confusing display.



In addition to giving you manual control over the appearance of your spreadsheet entries, the *Advantage* also allows the text attributes to be modified based on the results of calculations. Select cell B7. Click in the cell contents box past the end of the current formula. A cursor will appear at the end of the formula. Change the formula to read `"=B4-B5+( 0 * COLOR(B4-B5>=0,6,7) )"` and press RETURN. Now copy this new formula to all columns by selecting the range B7:H7 with the mouse and then selecting *Edit/Fill/Right*.

As the new formula is copied to each column, the result is calculated and displayed. This time, all of the net profit figures are displayed in black except for the figure for March. This number is negative and it is the only one displayed in red. This certainly flags out a problem month more clearly!

How does it work? The COLOR function is the key to this trick. It takes three arguments - a boolean (or logical) expression and two pen colors. The boolean expression is like a question that can be answered either "yes" or "no". In this case, the question is "B3-B4>=0"? In English, this is asking, "Is the net profit for the month (January) greater than or equal to zero?" If the answer to the question is yes, then COLOR will return the first pen color (6 = black). If the answer is no, COLOR will return the second pen color (7 = red). But in addition, COLOR will set the pen color for the cell where the formula appears, to the pen color returned. So when the profit for a particular month is less than zero, the calculated result will be displayed in red.

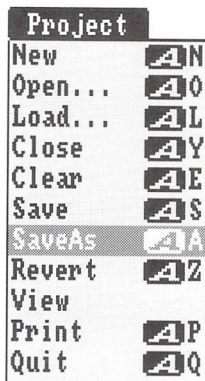
The value returned by COLOR is multiplied by zero before adding it to the net profit figure. This allows us to use COLOR to modify the text color without affecting the result of the calculation. The STYLE function allows you to modify the text attributes (italics, bold, underlined) in the same way.

Saving Your Work

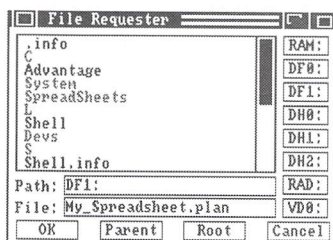
Now that our spreadsheet is complete, we should save a copy to disk. You will need a formatted disk for this operation. Place the disk in one

of your drives. Select *Project/SaveAs* from the menu.

The **File Requester** will appear. Select the *Drive Select* button, along the right side of the requester, for the drive containing your data disk. A list of the files currently on the disk will be displayed in the requester. Select the *File Name* box (currently displays "Untitled") and erase the contents of the box (LeftAmiga-X). Now enter a name for the spreadsheet ("My_Spreadsheet .ADV") and press RETURN. The *File* requester will close and your spreadsheet will be saved to the disk.



Click on the *Window Close* gadget in the upper left corner of your spreadsheet to remove it from the screen. You can recover your spreadsheet by following these directions:



- select *Project/Open*
- when the *File* requester appears, select the *Drive Select* button for the disk drive where you placed your data disk
- select the *File Name* box, type in "My_Spreadsheet.ADV" and press RETURN

A copy of the final version of the tutorial spreadsheet is stored in the "Examples" drawer on the the *Advantage* distribution disk. The spreadsheet is titled "Tutorial.ADV". You might want to load this spreadsheet to compare it to your work.

This concludes the introductory tutorial. You will want to read through the remainder of the manual to learn about the other features of the *Advantage*. First, you might want to take a look at the list of built-in functions and try experimenting with some formulas using these functions. As with most computer software, the best way to learn more about the *Advantage* is to use it!

CHARTS

Charts provide a concise and easy-to-understand means of conveying information. The charting functions provided by the *Advantage* allow you to present your data in a wide variety of basic formats. In addition, a number of options are provided for each type of chart, allowing you extensive control over the precise appearance of your charts. In this chapter we will cover the basic charting operations and examine the various options available.

Creating a Chart

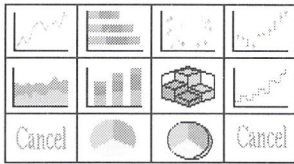
For the following examples, we'll make use of the spreadsheet "Scores.ADV" in the Examples drawer on the distribution disk.

To create a new chart you must first select the range of cells containing the data to be plotted. On the example spreadsheet this would include B5:F8. Labels to be displayed on the axes or in the chart legend should not be included in the selected ranges.

Each column (or, in some cases, groups of columns) in the selected range is considered to be comprised of a related group of values, called a "**data set**". In this example, each column represents the test results for a single student. All values in a given *data set* will be plotted in the same color.

A menu option is provided that allows you to select between data sets grouped by columns (the default) and data sets grouped by rows. Keep this in mind when reading the following instructions.

Commands	
Recalculate	A=
Goto Cell	AG
Cell Note	
Formula to Value	AB
Name	
User Formula	
New Chart	AI
Memory	AJ



To create a chart:

- Select the value cells to be plotted (B5:F8)
- Select *Commands/New Chart*
- The *Chart Type* requester will appear. This requester displays miniature representations of the available chart types. From left to right, top to bottom, the chart types are: line, bar, scatter, high/low bar, area, column, 3D column, step, pie, and 3D pie. Select the first chart type (line) by clicking on its image.
- The *Chart Options* requester will appear. This requester allows you to fine tune the appearance of your chart. For now, simply click on the *Continue* button to complete the chart.

When the chart window opens to display your new chart, you will see that each set of test scores (one per student) is displayed in a different color and line pattern. A legend, in the upper right corner of the chart, indicates which line represents each student's scores.

Chart Control	
Change Chart Type	AI
Prefs	AR
Set Crossing	AS
Use	
Set Max and Min	AH
Data	
AutoRedraw	
Redraw	

Chart Options Requester

A chart window provides different menu commands from those found in a standard spreadsheet window. Once you have a basic chart displayed, you can fine tune its appearance using these menu commands. Make sure that the chart window is selected (click anywhere within its bounds). Select *Chart Control/Prefs* from the menu. The *Chart Options* requester will open.

The *Chart Options* requester is the primary means of changing the appearance of a chart. The available options will vary depending on the current chart type. At the top of the requester is the Text Area, where up to two lines of text may be entered for the chart title (Header and Sub Header). Below the title lines are locations to enter labels to be placed on the x and y axes of the chart. Enter "Test Scores" as the Header, "Term" as the X Label, and "Score" as the Y Label.

Below the Text Area is the chart Options Area. This area provides a number of optional chart features. Click in the box to the left of the "Horizontal Grid" label. This option causes horizontal lines to be drawn across the chart at each tick mark along the y-axis. Finally, click on the *Continue* button to see the results of your changes.

Text	
Header:	
Sub Header:	
X Label:	
Y Label:	
Options	
☒ Ticks	☐ Horizontal Grid
Skip 0	☐ Vertical Grid
Start 1	☒ Horizontal Axis
☐ Values	☒ Vertical Axis
☐ Boxes	☐ Top Horiz Axis
☐ Drop Lines	☐ Right Vert Axis
☒ Legends	
Continue Cancel	

Other Format Controls

When a chart is initially displayed, the *Advantage* will automatically determine the range of values to be plotted along the y-axis. The range will be selected to encompass all of the data to be charted. You can adjust the range as desired by using the *Chart Control/Set Max and Min* command.

- Select *Chart Control/Set Max and Min*
- The *Chart Range* requester will appear. Fill in the desired maximum and minimum values to be plotted on the y-axis.
- Click on the *Continue* button.

Chart Control	
Change Chart Type	AT
Prefs	AR
Set Crossing	AS
Use	
Set Max and Min	AM
Data	
AutoRedraw	
Redraw	

By default, the *Advantage* will draw all bars in a chart from the axis outward. In some cases you may want the bars to be drawn outward from another point. For example, if you are displaying both positive and negative values, you might want the base of each bar to rest on zero, rather than the x-axis.

To draw a bar chart in this format:

- Select *Chart Control/Prefs*
- Activate the "Floating" option in the requester.
- Select *Chart Control/Set Crossing*
- The *Crossing Value* requester will appear.
Enter the y value at which you want the base of each bar to be situated.
- Click on the *Continue* button.

Chart Control	
Change Chart Type	AT
Prefs	AR
Set Crossing	AS
Use	
Set Max and Min	AM
Data	
AutoRedraw	
Redraw	

In addition, a number of commands are provided which allow you to specify the colors to use for drawing portions of the chart. The *Chart Control/Use* command lets you select between solid and patterned lines and area fills. The *Colors* menu lets you choose the color of the chart background, axes, titles and axis labels.

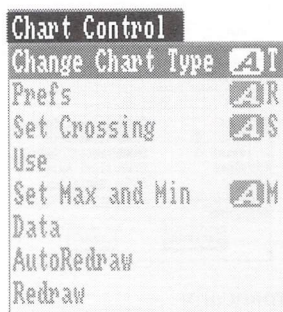


Chart Types

With the *Advantage*, you need not worry about choosing the correct chart format from the outset. If you don't like the appearance of a chart, you can try a different chart type at any time. To change the type of an existing chart:

- Select *Chart Control/Change Chart Type*
- The *Chart Type* requester will appear. Select the new chart format you wish to use.
- The *Chart Options* requester will appear. Select the desired format options and then click on the *Continue* button to display the chart.

TIP

For some chart types, you are expected to organize the data in a particular order on the spreadsheet. For scatter charts, the columns (or rows) in the range of cells to be plotted are grouped together in pairs. The left column (or top row) of each pair represents the x co-ordinate of the point. The right column provides the y co-ordinate. If an odd number of columns are selected for charting, the last column will not be plotted.

In high/low bar charts, the columns in the selected range are grouped by threes. The first column in each group indicates the low limit, the second column indicates the high limit and the final column provides the point value. For each point in a high/low bar chart, a vertical bar is drawn between the high and low limits with a tick mark displayed at the point value. This type of chart is useful for displaying such things as stock market trends, where you might display the high, low and closing values.

It is also useful for displaying experimental results, where it is usual to indicate error limits. The spreadsheet "Stock_Market.ADV" provides an example data set for charting in this format.

DATABASE OPERATIONS

The *Advantage* provides a number of commands designed specifically for database management. With these commands you can search, sort, extract and otherwise manipulate data stored on a spreadsheet. In addition, you can use the *Advantage's* other facilities to perform calculations based on the data and to chart any numeric data stored in your database.

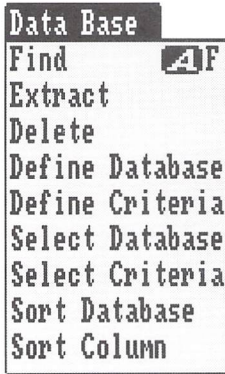
In the examples that follow, we'll make use of spreadsheet entitled "Employees.ADV" in the "Examples" drawer of the distribution disk. Load this spreadsheet into the *Advantage* now.

Defining a Database

Any range of cells within a spreadsheet can be defined as a database. The *Advantage* expects this range to be organized in the following manner:

- Each column in the range represents a **field** (a single item of data) within the database.
- Each row in the range, except for the first row, is a database **record** (a collection of related fields).
- The first row in the range is comprised of labels which indicate the **field names**. The field names in a database must be unique.

To define a database:



- Use extended selection to select the range of cells which comprise the database, including the field names and all records of the database.
- Select *DataBase/Define Database* from the menu.
- You will be prompted to enter a name for the database. Actually, you are providing a name for the selected range of cells, which happen to be organized in database format. Enter a name, up to 15 characters long, using only alphabetic characters, periods (".") and underscores ("_").
- If the name is acceptable, the **Advantage** will indicate that the named range has been added to the list. The selected range becomes the current database.

Try this now with the Employees spreadsheet. Select range A4:F11 and define it as a database with the name "DB_EMPLOYEE".

TIP

Since the **Advantage** does not differentiate between simple named ranges and those used for databases and criteria, you might want to start all database names with a "DB_" and criteria names with a "CR_" so that you will be able to readily identify them.

Sorting a Database

Once you have selected a database, you can sort it on the first field. Select *DataBase/SortDatabase* and you will see that the records of DB_EMPLOYEE are reorganized so that the first field, "Sort Key", is arranged in order from the smallest to largest value.

TIP

Since a database can only be sorted based on its first field, you may want to make use of the trick that we have used here, in your own databases. The "Sort Key" field of DB_EMPLOYEE is a calculated field. It uses the CELL() built-in function to extract the field at a given offset from the start of each record. And the offset used is extracted from cell B2, labelled "Key Column". This trick makes it a simple matter to sort your database based on any numeric field (CELL() returns numeric values only).

Try this:

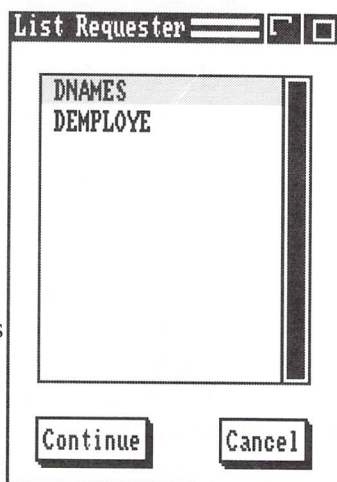
- Select B2 and change it to 2, to select the Age field as the sort key.
- Click on the *Calc* button to update the Sort Key field of each record.
- Select *Data Base/Sort Database* to sort the records based on employee age.

Multiple Databases

You can, of course, use a separate spreadsheet for each database you want to maintain. If you have several databases containing related information, you might want to store them on a single spreadsheet. The *Advantage* allows you to do this. In fact, you can even have one database that is contained within another. One application of this can be seen in our example database.

Select the range B4:F11 and define it as database "DB_NAMES". The first field of this database is the employee name. Now select *Data Base/Sort Database* and the records will be sorted by employee name. You can bring the Sort Key field back into synch with the record beside it, if desired, by clicking on the *Calc* button.

The database commands operate on the current (selected) database. When you have multiple databases stored on a single spreadsheet, you can quickly make any one of them current. Select *Data Base/SelectDatabase* from the menu. The *List requester* will appear. This requester displays all named ranges on the current spreadsheet (this is where the TIP about range names comes in handy). Click on "DB_EMPLOYEE" and then click on the *Continue* button. DB_EMPLOYEE is now the current database. Select *Data Base/Sort Database* to sort the records and convince yourself that this is true.



Selection Criteria

The remaining database operations all operate in conjunction with **selection criteria**. Selection criteria are a means of specifying a subset of the records in your database. They define one or more conditions

that each record in the subset must meet.

The format of a criterion is:

- A row of field names indicating the fields of each record that will be tested. A field name may appear more than once if multiple tests are to be performed on the field contents.
- One or more rows of formulas. These formulas are situated in the same column as the name of the field to which they apply. Each formula indicates a test to be performed on the contents of the associated field.
- An empty cell will match all field entries.

If the formula is a constant (ex/ 23) or a regular formula (ex/ = A2 * C2) then the field contents must exactly match the criteria value for the record to be considered a match. More flexible tests may be made using special **criteria formulas**. A criteria formula begins, as usual, with an equals sign. Unlike a regular formula, however, it includes a logical operator and then a value or formula for comparison purposes. (Refer to the definition of a "Boolean Expression" in the *Functions* section for a complete list of logical operators.)

One example of a selection criterion is located in cells B14:C15 of the Employee spreadsheet. This criterion (named "CR_AGELIMIT") consists of two columns, both labelled "Age". Both tests of this criterion are made on the Age field of the database records. The formulas for this criterion are both criteria formulas. The first one tests whether the Age for the record is "greater than or equal to the value stored in cell B16". The second test for an Age "less than or equal to the value stored in cell C16. We could have used constants here (ex/ = >=25) but this example demonstrates that the comparison value can be a calculated value if you wish.

For a given row of a selection criterion, the tests in each column are AND'ed together. This means that all comparisons in the row must be TRUE for a given record in order for the record to be selected. If the selection criterion consists of more than one row, the rows are OR'ed together. This means that a record will be selected if it matches any of the rows of the selection criterion.

Fields containing text may be selected using string patterns. Such patterns may contain asterisk (*) and question mark (?) wild cards. An asterisk will match zero or more characters. A question mark matches any single character. If the pattern contains no wild cards, it must exactly match the

field entry. For example:

= "bend" would match "bend"

= "b?nd" would match "bend", "band", "bond", etc.

= "b*nd" would match "bend", "bnd", "bound", etc.

Searching a Database

The *Find* command in the *Data Base* menu lets you search for records in the current database that match the current selection criterion. A number of selection criteria are already defined on the example spreadsheet. Choose the age limit criteria as follows:

- Select *Data Base/Select Criteria*
- The *List* requester will open. Select CR_AGELIMIT from the list.
- Click on the *Continue* button.

Now select *Data Base/Find*. The first record in the Employee database with an age between 25 and 30 will be highlighted. You can search for succeeding records that match the criteria by selecting *Data Base/Find* again (or use the keyboard shortcut, RightAmiga-F or Shift-F1). The message "I couldn't find any more records." will be displayed when no more matching records can be found.

Extracting Records

The *Extract* command is used to make copies of records which match the current selection criteria.

- Select the database from which you wish to extract records
- Select the selection criteria which will define the records to be extracted
- Select the location within the work area where you want to place the copies of the extracted records. Move the **cell cursor** to the upper left corner of this destination range. Make sure you have enough space for the extracted records. They will overwrite any existing data.
- Select *Data Base/Extract*

Deleting Records

The *Delete* command is used to remove records which match the current selection criteria.

- Select the database from which you wish to delete records
- Choose the selection criteria which will define the records to be deleted
- Select *Data Base/Delete*

All matching records will be deleted and the remaining records will be compressed onto consecutive rows on the spreadsheet. A delete operation cannot be undone, so be careful. You might want to use the *Find* or *Extract* command before deleting records to ensure that the desired records are the only ones which will be discarded.

Database Functions

The *Advantage* provides a number of built-in functions specifically designed for use with databases. These functions allow you to perform a calculation involving a subset of your database records by including a selection criteria in the function argument list.

Take a look at cell G13 on the example spreadsheet. This cell displays the average income of employees whose age falls within the limits indicated in the age limit criterion, CR_AGE LIMIT. (Note: this cell will display "ERROR" until the spreadsheet has been recalculated with the DB_EMPLOYEE database defined.) Try changing the age limits and then recalculating the spreadsheet.

MACROS

The *Advantage* provides a macro facility that allows you to record and play back a series of commands. If you find that you frequently repeat a series of keystrokes or menu commands, you will find this capability very convenient. In addition, macros allow you to extend the capabilities of the *Advantage's* ARexx interface since any macro may be executed from an ARexx program. For example, there is no direct ARexx command provided to execute any of the *Advantage's* menu commands. However, a macro can be created to execute the desired command and this macro can, in turn, be called from an ARexx program.

There are certain limitations placed on macros. These include:

- macros operate only on the current spreadsheet window (they can't be used to cut data from one spreadsheet window and paste it into another, for example)
- macros don't record changes to windows such as window sizing, depth arrangement and dragging (repositioning)

Creating a Macro

A new macro is created by placing the *Advantage* in a "learn" mode and simply executing the sequence of keystrokes and commands you wish to record.

Macros

```
Open Macros
Save Macros
Close Macro
Close ALL Macros
Start Recording...
End Recording
Play Back!
Arexx
```

- Select *Macros/Start Recording*. The window title will change to indicate that you are in macro recording mode.
- Perform the desired sequence of keystrokes and/or menu commands just as you would normally execute them.
- Select *Macros/End Recording*
- You will be prompted for a macro name. Enter a unique name of up to 18 characters to complete the macro. Press RETURN without entering a name to cancel creation of the macro.
- If you wish to cancel the macro while you are in macro record mode, press the ESC key.

All movements of the *cell cursor* in the macro are recorded as relative to the cell that was selected when the macro recording was begun. This allows macros to be played back at any location on the spreadsheet. For example, you could record a macro that enters the month names into a row of cells starting at A1 and ending at L1. If you then selected cell C3 and played this macro back, the month names would be entered into cells C3 through N3.

Playing a Macro

The *Play Back* command is used to execute an existing macro.

- Select the cell where you would like the macro sequence to begin (not all macros will write data into the work area, so this step may not be required).
- Select *Macros/Play Back*. The *List* requester will open and display the names of all memory resident macros.
- Select the desired macro
- Click on the *Continue* button

Assigning a Macro to a Key

To speed up access to macros, the *Advantage* allows macros to be accessed by a key. To assign a macro to a key, press and hold the CTRL key then the key you wish to assign the macro to (eg "M"). The list requester will open and display the names of all memory resident macros. Pick one (or select the cancel button to abort the operation). The next time that the CTRL-M combination is pressed, the macro assigned to it will be executed.

NOTE: All qualifiers, like the shift, alt, and Amiga keys are ignored. The numeric keypad keys are considered different from the keys on the keyboard.

Attaching Arexx Scripts

The *Advantage* allows you to attach Arexx scripts to individual cells within the spreadsheet. These programs are run by double clicking on the cell to which they are attached. This is a useful and a powerful tool that gives your spreadsheet the power to communicate with other programs. For example, on a spreadsheet detailing the rise and fall of value in a portfolio of stocks, a cell with the text update in it could have an ARExx script attached to it. Double-clicking on this cell would run the script which, in this case, would run a telecommunications program to access, via modem, a stockbroker network. Highs and lows, for selected stocks in the spreadsheet, would be automatically updated, via ARExx. To attach an ARExx script to a cell:

- Select the cell to which you wish to attach the script.
- Select *Macros/ARExx/Attach*. This will bring up the *File* requester with which you would select any one of your current ARExx scripts to attach to the selected cell.

The script will be executed whenever you double-click on the assigned cell.

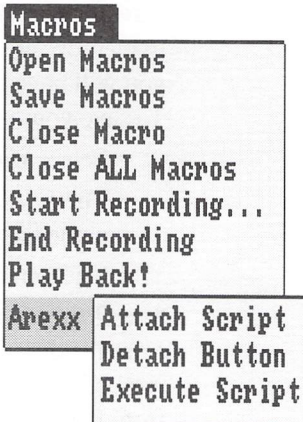
Detaching an Arexx Script

- Select cell from which you want to remove the script
- Select *Macros/ARExx/Detach*. If no script is attached to the currently selected cell this option will have no effect.

Executing Scripts

The *Advantage* also allows you to execute Arexx scripts without attaching them to cells in your spreadsheet.

- Select *Macros/ARExx/Execute* to bring up the *File* requester with which you can choose an ARExx script
- Select an ARExx script. It will now run as if attached to a cell in your spreadsheet and double-clicked.



Other Macro Commands

Once you have created a number of macros, you will want to save them to disk so that you can use them in your next session with the *Advantage*. When you save your macros, ALL of the macros currently in memory are saved together. If you wish to save individual macros, you will have to clear the undesired macros from memory (see below). To save your macros:

- Select *Macros/Save Macros*
- The *File* requester will open. Enter or select the directory and file name for your macros.
- Click on the *OK* button

The *Open Macros* command allows you to load macro definitions from the disk. These macros will be appended to the list of macros currently in memory. Note that the *Advantage* will not prevent you from loading macros with names which are identical to those that are already in memory. To avoid confusion, try to use unique names for each of your macros.

The remaining commands, *Close Macro* and *Close ALL Macros*, are used to remove macro definitions from memory. The first command allows you to selectively remove macros, one at a time, by choosing them from the *List* requester. The second command provides a quick method of clearing all existing macros.

The "Macros" drawer on your distribution disk contains a number of example macros.

AREXX

If you have purchased a copy of the ARExx language, this chapter will be of interest to you. The *Advantage* provides an ARExx interface and special command language with which ARExx macros can be constructed. This command language provides a powerful extension to the *Advantage* which allows you to build complex operations that are not directly available in the program.

In order to use these commands, the *Advantage's* ARExx port must be activated. To do this, select the "ARExx On" option in the *Advantage's* *Preferences* requester.

You will have to execute your ARExx macros from a CLI window on the Workbench since the *Advantage* does not provide a method of doing so from within the program.

Note the following points when creating ARExx macros for the *Advantage*:

- The host address for the *Advantage* is "Advantage".
- Command parameters must each be passed on a separate line, as shown in the examples below. Leading blanks are not necessary but help with program clarity.
- Commands will return a status code in the special ARExx variable RC. A return code of 0 indicates successful completion,

40 indicates a syntax error (possibly you misspelled the command), and 100 indicates that the ARexx interface is not active (check the *Preferences* settings).

- The command "options results" must appear in your program prior to the first use of any commands that return results, in order for such commands to work properly.
- Commands which return results, return them as strings in the special ARexx variable RESULT (as long as the command was successfully completed).

ARexx Commands

CloseAllMacros - Closes all the macros in memory. Probably should be used before LoadMacro to eliminate macro name conflicts. CloseAllMacros doesn't need any parameters.

Current - Returns the currently selected cell or cell range.

DrawMessage - Displays a message in the *message box*.

Example: `DrawMessage`
 "This is the message"

GetFormula - Returns the formula from the current cell (if the current cell contains a formula).

GetLabel - Returns the label from the current cell (if the cell contains a label).

GetString - Allows the user to get a string from the data entry box on the window. This function should facilitate data entry.

Example: `GetString`
 "Message to go in Message Box"
 "Initial String"

GetValue - Returns the value from the current cell (if the cell contains a formula or value). The result string returned will be exactly what is displayed on the screen. If the current cell contains a label, a null string is returned.

IsFormula - Returns 1 if the current cell is a formula cell. Otherwise, returns 0.

IsLabel - Returns 1 if the current cell is a label cell. Otherwise, returns 0.

IsValue - Returns 1 if the current cell is a value cell. Otherwise, returns 0.

LoadMacro - Loads in a macro file from disk.

Example: *LoadMacro*
 "FileName"

LoadSpread - Loads in a spreadsheet from disk. Clears any data in the current window.

Example: *LoadSpread*
 "Filename"

Macro - Executes the specified macro (which must already be loaded into memory).

Example: *Macro*
 "MyMacro"

PutCell - Puts an entry into the current cell as if the user had typed it from the keyboard. The entry must be enclosed in quotes to be processed properly.

Example: *PutCell*
 "= A1 + B2"

SaveSpread - Saves a spreadsheet already in memory to disk.

Example: *SaveSpread*
 "FileName"

SelectCell - Makes the specified cell the current cell.

Example: *SelectCell*
 A1

SelectRange - Selects the range of cells delimited by the top left corner and bottom right corner. The top left corner becomes the current cell.

Example: *SelectRange*
 A1
 B6

MENU REFERENCE

This chapter provides a quick reference to all of the menu commands available in the *Advantage*. Keyboard shortcuts for the commands are indicated, where available, in brackets following the command name. The conventions used here are:

RightAmiga-x	Hold the right AMIGA key down and press key "x"
Shift-x	Hold the SHIFT key down and press key "x"
Alt-x	Hold the ALT key down and press key "x"
Ctrl- x	Hold the CTRL key down and press key "x"

Note that some commands have more than one keyboard shortcut.

A number of keyboard shortcuts are available using the function keys (labelled F1 through F10 on your keyboard). A list of the shortcuts assigned to these keys may be viewed by pressing the HELP key.

TIP

Project Menu

New (RightAmiga-N)

Opens a new (blank) spreadsheet window.

Open (RightAmiga-O)

Opens a new spreadsheet window and loads an existing *Advantage*, *Maxiplan* or *Lotus 1-2-3* spreadsheet into this window.

Project	
New	 N
Open...	 O
Load...	 L
Close	 Y
Clear	 E
Save	 S
SaveAs	 A
Revert	 Z
View	
Print	 P
Quit	 Q

Load (RightAmiga-L)

Loads an existing spreadsheet into the current spreadsheet window, overwriting any data currently stored there.

Close (RightAmiga-Y)

Closes the current spreadsheet window and releases all memory allocated for the window contents. A spreadsheet window may also be closed by clicking on its *window close* gadget.

Clear (RightAmiga-E)

Erases all data in the current spreadsheet window and releases the memory allocated to store this data.

Save (RightAmiga-S or F9)

Saves the current spreadsheet under the same name it was last saved under (or as "Untitled", if the spreadsheet has not been saved before). It is a good idea to save your work

frequently to avoid a possible loss of work due to a power failure or system crash.

SaveAs (RightAmiga-A)

Saves the current spreadsheet to disk as an **Advantage** or Lotus 1-2-3 document, under a new file name. Once a spreadsheet has been saved, its new file name will appear in the window title bar.

Revert (RightAmiga-Z)

Restores the spreadsheet from the last copy saved to disk. This operation overwrites any data currently on the spreadsheet.

View

Opens a secondary window for the current spreadsheet. This window will display the same data that is stored on the current spreadsheet. It can be scrolled independently of the original window, allowing you to simultaneously view two separate regions of the spreadsheet. You may open as many *View* windows as memory will allow. Changes made in any view will immediately be reflected in all other views of the same spreadsheet.

Print (RightAmiga-P)

Prints the selected range of cells.

- Select the range of cells to be printed (use *Edit/Select All* to print the entire spreadsheet)
- Select *Project/Print*. The *Print* requester will open.
- Select any of the following print options:

Header: text to print at the top of each page

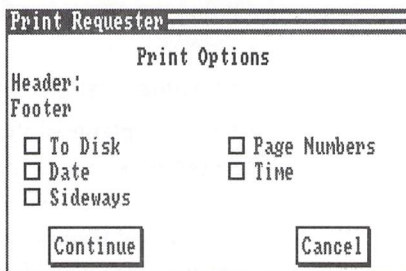
Footer: text to print at the bottom of each page

Date: print the date at the top of each page

Time: print the time at the top of each page

Page Numbers: print a page number at the bottom of each page

To Disk: output the report to a disk file. This file may be edited with a text editor and printed by copying it to the PRT: device. The *File* requester will appear after you have selected *Continue*, to allow you to choose the destination file.



Sideways: print the report sideways on the paper. Use this option if the range of cells you wish to print is too wide to fit on the page in the normal orientation.

- Click on the *Continue* button to start the printout
- Click on the *Cancel* button to abort the printout

The printout will contain the information currently displayed on the spreadsheet. If you wish to print the spreadsheet formulas, choose the Formulas option in the *Preferences* requester before printing. Use the Preferences tool on your Workbench disk to configure your printer as desired before printing (page size, margins, etc.).

Quit (RightAmiga-Q)

Exits from the *Advantage* after asking for confirmation.

Edit Menu

Cut (RightAmiga-X)

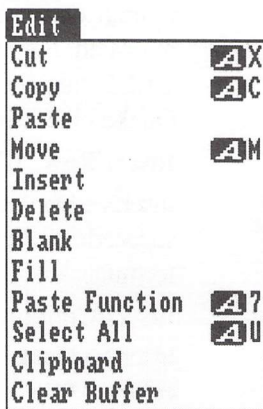
Copies the contents of the currently selected range of cells to a temporary work space and then blanks the selected range.

Copy (RightAmiga-C)

Copies the contents of the currently selected range of cells to a temporary work space. The selected cells are left unchanged.

Paste/Full (RightAmiga-V)

Pastes the contents of the temporary work space to



the spreadsheet. The range of cells in the work space is positioned such that the upper-left cell of the range goes into the current cell on the spreadsheet.

Paste/Values (RightAmiga-I)

Pastes the contents of the temporary work space to the spreadsheet. This command works like the *Paste/Full* command except that all formula cells in the work space are reduced to value cells (by computing the result of each formula) before completing the paste operation. The values used in the computations are those that were in effect at the time the cell contents were cut or copied to the work space. This command is useful for replacing a table of computed values with the equivalent constants, in cases where the table needs to be computed only once. This allows you to reduce both the recalculation time and storage space required for the spreadsheet. See also *Commands/Formula to Value*, later in this section.

Paste/Relative

Pastes the contents of the temporary work space to the spreadsheet, much the same as with *Paste/Full*, but all cell references are changed so that the relative offset from the destination cell remains constant

Move (RightAmiga-M)

Pastes the contents of the temporary work space to the spreadsheet. In addition, *Move* modifies all cell and cell range references so that the formulas are properly relocated to the new location. Only cell references entirely contained within the original copied range will be relocated. For example, if you cut the range A1:B2 and moved it to C1:D2, then the formula " $=A1+B2$ " would be changed to " $=C1+D2$ ". On the other hand, " $=SUM(A1:D7)$ " would be left unchanged.

Insert/Row (RightAmiga-0)

Inserts a blank row into the spreadsheet before the current row. All succeeding rows of the spreadsheet are pushed down one row. All formulas are adjusted to account for the insertion of the new row.

Insert/Column (RightAmiga-1)

Inserts a blank column into the spreadsheet to the left of the current column. All succeeding columns of the spreadsheet are pushed right one row. All formulas are adjusted to account for the insertion of the new column.

Delete/Row (RightAmiga-2 or F6)

Deletes the current row and moves all succeeding rows of the spreadsheet up one row. All formulas are adjusted to account for the removal of the row.

Delete/Column (RightAmiga-3 or F7)

Deletes the current column and moves all succeeding columns of the spreadsheet left one row. All formulas are adjusted to account for the removal of the column.

Blank/Row (RightAmiga-4)

Blanks out all cells in the current row and releases the memory used for these cells back to the system.

Blank/Column (RightAmiga-5)

Blanks out all cells in the current column and releases the memory used for these cells back to the system.

Blank/Range (RightAmiga-6 or F8)

Blanks out all of the currently selected cells and releases the memory used for these cells back to the system.

Fill/Down (RightAmiga-D or F2)

Copies cell contents from the top row of the selected cells to succeeding rows. Any cell references in the copies are adjusted so that the relative offset from the destination cell remains constant.

Fill/Right (RightAmiga-R or F3)

Copies cell contents from the leftmost column of the selected cells to succeeding columns. Any cell references in the copies are adjusted so that the relative offset from the destination cell remains constant.

Paste Function (RightAmiga-7)

Allows a reference to a built-in function to be pasted into a formula cell.

- Select the cell to be modified
- Enter the leading equals sign (=) or click in the *cell contents box* where you wish to insert the function reference
- Select *Edit/Paste Function*
- When the *List requester* opens, select the name of the function to

be inserted

- Click the *Continue* button

The function name and dummy argument list will be inserted into the cell contents box. You should edit the argument list to replace the dummy values with actual arguments. The type of argument expected is indicated by the dummy argument according to the following table:

V = a value

R = a range

C = a cell reference

S = a string

L = a list

Repeat the above procedure to paste further functions into the formula.

Select All (RightAmiga-U)

Selects a range which encompasses all non-blank cells on the spreadsheet. This command is useful for making global format changes. For example, to change all values displayed on the spreadsheet to a common format; select this command and then select the new numeric format.

Clipboard/Copy Data

Copies the selected range of cells to the Amiga's clipboard. This command copies the information that is currently displayed on the screen to the clipboard as simple text, formatted in much the same way as it appears on the screen. Each row of the copy's range is terminated by a line feed character. This command allows data to be exported to other programs which support the clipboard device.

Clipboard/Paste Data

Pastes text from the Amiga's clipboard into successive cells of the spreadsheet, starting from the current cell. A new cell entry is started each time one or more spaces are encountered in the text. A new row is started each time a line feed character is encountered. This command allows data to be imported from a program which properly supports the clipboard device.

Clear Buffer

Clears the contents of the temporary work space and releases the memory back to the system for other uses.

Format Menu

General (RightAmiga-#)

Selects fixed or floating point format (ex/ 4.5) for display of numbers in the currently selected cells. This command only affects cells containing numeric values. The choice between fixed and floating point format is determined by the *Fixed Decimals* command discussed on the following page.

Scientific (RightAmiga-H)

Selects scientific notation format for display of numbers in the currently selected cells. This command only affects cells containing numeric values. For example, the value 34000 would be displayed as "+3.4E+04".

Percent (RightAmiga-%)

Selects percent format for display of numbers in the currently selected cells. This command only affects cells containing numeric values. For example, the value 0.15 would be displayed as "15%".

Currency (RightAmiga-\$)

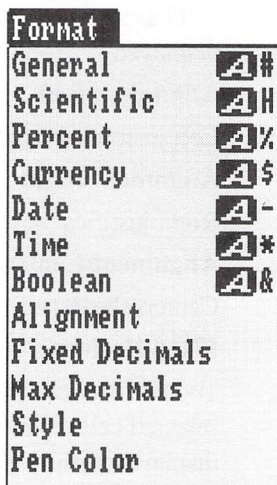
Selects currency format for display of numbers in the currently selected cells. This command only affects cells containing numeric values. Values are displayed in fixed point format to two decimal places with a leading dollar sign (ex/ "\$10.25").

Date (RightAmiga--)

Displays values in the currently selected cells in date format (ex/ "Jan-01-78"). The value in each cell is interpreted as the number of days since January 1st, 1978 (only the integer portion of the value is used).

Time (RightAmiga-*)

Displays values in the currently selected cells in time format (ex/ "01:26PM"). The value in each cell is interpreted as a fraction of a day



(only the fractional portion of the value is used). Zero (0) is midnight and 0.9999 is 11:59 pm.

Boolean (RightAmiga-&)

Displays values in the currently selected cells in boolean format. If a cell has a value of zero, "FALSE" is displayed. Otherwise, "TRUE" is displayed.

Alignment/Left

Left justifies the data displayed in the selected cells.

Alignment/Right

Right justifies the data displayed in the selected cells.

Alignment/Centered

Centers the data displayed in the selected cells.

Fixed Decimals

Toggles between fixed and floating point numeric format in the selected cells. When *Fixed Decimals* is active, the number of digits displayed to the right of the decimal point is determined by the *Max Decimals* command. This option may be used in conjunction with cells in the *General*, *Scientific* or *Percent* formats.

Max Decimals/0..14

Sets the number of digits to be displayed to the right of the decimal point in fixed point numbers. This command operates on the currently selected cells.

Style/Plain

Displays text in the selected cells in the normal font, removing any italics, bolding and underlining.

Style/Italic

Displays text in the selected cells in italics. If the text is already displayed in italics, selecting this removes the italic attribute.

Style/Bold

Displays text in the selected cells in bold letters. If the text is already bold, selecting this removes the bold attribute.

Style/Underline

Underlines the text in the selected cells. If the text is already

underlined, selecting this removes the underline.

Pen Color

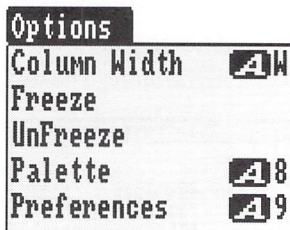
Chooses the pen color for text in the currently selected cells.

Options Menu

Column Width (RightAmiga-W)

Sets the width of the currently selected columns.

- Choose columns by selecting any cells which span the columns to be affected
- Select *Options/Column Width*
- Enter the new column width
- Press RETURN



Freeze/Row

Lock the current row into the top of the work area. This row will remain in place when the work area is scrolled. Only one row may be frozen at any given time.

Freeze/Column

Lock the current column to the left side of the work area. This column will remain in place when the work area is scrolled. Only one column may be frozen at any given time.

UnFreeze/Row

Unlock the row currently frozen at the top of the work area.

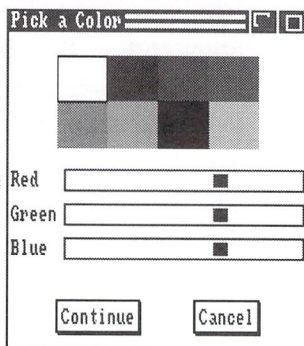
UnFreeze/Column

Unlock the column currently frozen at the left side of the work area.

Palette (RightAmiga-8)

Opens the *Palette* requester so that you may modify the display colors.

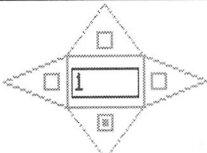
- Select the color you wish to modify from the color samples at the top of the requester.
- Adjust the Red, Green and Blue color composition sliders to change the color as desired.



- Click on the *Continue* button to close the Palette Requester and use the modified palette.
- Click on the *Cancel* button to close the Palette Requester and without making any changes.

Preferences (RightAmiga-9 or F4)

Allows you to view and modify the preference settings for the *Advantage*. The available options include:

Preferences Requester		
Calculate ☒ ByRow ☐ ByColumn ☒ Manual ☐ Automatic	Show ☒ Values ☐ Formulas ☒ Hollow Box ☐ Filled Box	Max Rows 100 Max Cols 100 Iterations 1
Grid ☐ On ☒ Off	WorkBench ☐ On ☒ Off	Expert ☐ On ☒ Off
		Continue Cancel Reset
		Say ☐ On ☒ Off
		ARexx ☐ On ☒ Off

Calculate ByRow/ByColumn

Chooses between recalculating the spreadsheet from the top row to the bottom row, one row at a time (ByRow), or from the leftmost column to the rightmost column, one column at a time (ByColumn). Click in the hollow box to the left of the desired calculation order. A solid box will appear in the currently selected option.

Calculate Manual/Automatic

When Manual is selected, you must click on the *Calc* button or select *Commands/Recalculate* to recalculate the spreadsheet once you have made any changes. In Automatic mode (the default), the spreadsheet is recalculated each time you make a change.

Show Values/Formulas

Chooses between displaying the results of calculations (Values) or the formulas used to calculate each cell (Formulas).

Show Hollow Box/Filled Box

Chooses between a border (Hollow Box) or a solid area (Filled Box) to indicate the current location of the *cell cursor*.

Max Rows and Max Cols

Sets the dimensions of the spreadsheet work area. To change a dimension, click in the value box, erase the existing value, enter the

desired number (up to 65000) and press RETURN. Note that empty rows and columns do consume a small amount of memory. You should try to set these values close to your actual requirements in order to conserve memory.

Iterations

Sets the number of times that the entire spreadsheet is recalculated each time an automatic or manual recalculation is performed. By setting this value higher than 1, it is possible to have the spreadsheet properly calculated even if a cell calculated early in the sequence depends upon the value of a cell calculated later on. Change this value as described above, for Max Rows and Cols.

Grid On/Off

Displays or hides grid lines that mark out the individual cells in the spreadsheet work area.

WorkBench On/Off

Opens (On) or closes (Off) the Workbench screen. Closing the Workbench screen makes an extra 30K of memory available to the *Advantage*. The *Workbench* screen will not close if any application programs (including the *Advantage* itself) are making use of the screen.

Expert On/Off

Disables many of the warning type requesters such as those the *Advantage* uses to remind you to save your work before quitting.

Say On/Off

When *Say* is On, each cell entry you complete will be read back to you.

ARexx On/Off

Turns the ARexx interface on or off.

Auto Advance Options

These gadgets, located in the bottom left corner of the *Preferences* requester, allow you to select the direction and amount by which the *cell cursor* will advance each time you complete a cell entry. Click on one of the arrows to select the auto advance direction. The current direction is indicated by a solid box in the arrow. Select the value box, in the center of the direction arrows, and enter the number of cells by which the *cell cursor* should advance. If you want the *cell cursor* to

remain stationary, enter a zero (0) into this box. Press RETURN to complete the entry.

Select the *Reset* button to restore the default preference settings. Select *Continue* to close the Preference Requester and use the new preference settings. Select *Cancel* to close the requester and abort any changes you have made. A set of preferences settings are saved and restored with each spreadsheet.

Commands Menu

Commands	
Recalculate	 =
Goto Cell	 G
Cell Note	
Formula to Value	 B
Name	
User Formula	
New Chart	 T
Memory	 J

Recalculate (RightAmiga-= or F10)

Recalculates and displays all cells on the spreadsheet which require updating.

Goto Cell (RightAmiga-G or F5)

Moves directly to any specified cell on the spreadsheet. The indicated cell is displayed in the upper-left corner of the work area.

Cell Note/Create

Allows a cell note to be entered for the current cell or displays the existing cell note and allows you to edit it. A cell note is any text you wish to attach to a particular spreadsheet cell - possibly a help message or explanatory note.

Cell Note/Create Extended

Allows you to create or read a text file, using *TransWrite*, which is attached to the cell note. Whereas a normal cell note allows you to display one line of text with information pertaining to a particular cell, with the extended cell note you can go into a more lengthy description of a cell, a section of a spreadsheet or an entire spreadsheet. In order to use the extended cell note feature of the *Advantage* it is necessary to have a copy of *TransWrite* running in the background. After you have started the *Advantage*, use the screen *front-to-back* gadget, located in the upper right hand corner of the the *Advantage* screen, to bring forward the *Workbench* screen. If you are unclear on the use of screen front-to-back gadgets, they are fully explained in the documentation provided with your Amiga. Now, place your *TransWrite* disk in a disk drive and double-click on the *TransWrite* icon. A *TransWrite* window, available for extended cell notes, will open on the *Workbench* screen. To create an extended cell note:

- Select the cell in your spreadsheet to which you want the note attached.
- Select *Commands/Cell Note/Create Extended*. This will bring up a **TransWrite** window which will allow you to type and edit a text file of any length. When you have completed the text for your extended cell note select *Special/Send Text Home* from **TransWrite's** *Special* menu. This sends your complete extended cell note to the **Advantage** where it is stored as part of the spreadsheet. To view this text, double-click on the cell note box in the **Advantage** wherever you see the message "Extended Cell Note". This will bring up a **TransWrite** window showing the full cell note and allowing you to edit it.

Cell Note/Delete

Removes the cell note from the current cell.

Formula to Value (RightAmiga-B)

Converts all formula cells in the selected range to value cells. This command is useful for replacing a table of computed values with the equivalent constants, in cases where the table needs to be computed only once. This allows you to reduce both the recalculation time and storage space required for the spreadsheet. See also *Edit/Paste/Values*.

Name/New

Adds a named cell or range to the name list.

- Select the cell or range to be added
- Select *Commands/Name/New*
- Enter the name. It may be up to fifteen (15) characters in length and must be composed of upper case letters (A-Z), period (.) and underscore (_) only.
- Press RETURN

A named cell or range may be used anywhere that a standard cell name or cell range is acceptable. This allows you to write self-documenting formulas, such as " = TAXRATE * PRICE " to calculate the amount of tax on an item. Also, note that names are absolute references. They are not affected by *Move*, *Insert*, *Delete* and *Fill* operations.

Name/Change

Changes the cell or range of cells to which an existing name refers.

- Select *Commands/Name/Change*
- When the *List* requester opens, select the name to be redefined
- Click the *Continue* button

When the spreadsheet is recalculated, any cells referring to the modified name will be updated appropriately.

Name/Clear

Removes a named cell or cell range from the list of names.

- Select *Commands/Name/Clear*
- When the *List* requester opens, select the name to be removed
- Click the *Continue* button

TIP

Note that since databases and selection criteria are simply special kinds of named ranges, they will also appear in the *List* requester. Therefore, you can remove a database or criteria range by using *Commands/Name/Clear*.

Name/Select Name

This will bring up the list requester showing all of the available user-defined names. If one of these names is selected, then the cell or range that the name represents will be selected.

User Formula/New

Adds a user defined formula to the formula list.

- Select the cell containing the formula to be added to the formula list
- Select *Commands/User Formula/New*
- Enter the formula name. It may be up to eight (8) characters in length and must be composed of upper case letters (A-Z) only.
- Press RETURN

User formulas are "called" by including the formula name, followed by a pair of round brackets, in a formula cell. For example, the user formula "MYFORM" might be used in a formula such as "=A1 * MYFORM()". Since user formulas do not support argument lists, it is common to use **relative** cell and range references in the formula. In this way, the arguments for the formula will be dependent on the location of the cell containing the formula call. The built-in functions

CELL() and RANGE(), described in the *Functions* section, are used to make relative cell references.

Once a user formula has been defined, the original formula cell may be modified or even deleted from the spreadsheet. The user formula is retained separately from the spreadsheet entries so that it is not affected.

User Formula/Change

Redefines the formula associated with an existing user formula name.

- Select the cell containing the new formula
- Select *Commands/User Formula/Change*
- When the *List* requester opens, select the formula name to be redefined
- Click the *Continue* button

When the spreadsheet is recalculated, any cells referring to the modified formula will be updated to reflect the change. This is one of the advantages of employing user formulas - the formula need only be changed in one place to correctly modify all references to it.

User Formula/Clear

Removes a user formula from the formula list.

- Select *Commands/User Formula/Clear*
- When the *List* requester opens, select the name of the formula to be removed
- Click the *Continue* button

Any formulas which refer to the discarded user formula will display the message "ERROR" when the spreadsheet is next recalculated. You should remove all references to the discarded formula from the spreadsheet.

New Chart (RightAmiga-T)

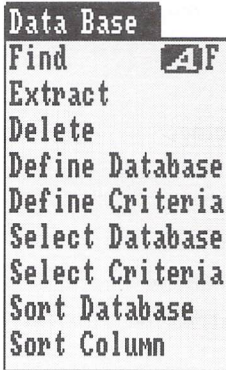
Initiates creation of a new chart incorporating data from the currently selected cells on the spreadsheet.

- Select the range of cells containing the data to be charted. Include only the data cells, not the text to appear on the x-axis or in the legend.
- Select *Commands/New Chart*
- The *Chart Selection* requester will open. Select the desired type of chart.

- The *Chart Format* requester will open. Fill in any desired chart text and select the desired chart format options.
- Click on the *Continue* button and your data will be charted.

Memory (RightAmiga-J)

Displays a message indicating the amount of memory available (Chip, Fast and Total).



Data Base Menu

Find (RightAmiga-F or Shift-F1)

Finds the next record in the current database that matches the current selection criteria. If a matching record is found, it is highlighted. Otherwise, a message is displayed indicating that no more matches were found.

Extract (Shift-F2)

Extracts all records in the current database that match the current selection criteria. Makes copies of these records at a new location on the spreadsheet, starting at the current cell.

Delete (Shift-F3)

Removes all records in the current database that match the current selection criteria. The remaining records are compressed onto consecutive rows on the spreadsheet.

Define Database (Shift-F4)

Defines a named range on the spreadsheet and selects it as the current database. The first row of the range should consist of field names. The succeeding rows comprise the database records.

Define Criteria (Shift-F5)

Defines a named range on the spreadsheet and selects it as the current selection criteria. The first row of the range should consist of field names. Succeeding rows define the selection criteria.

Select Database (Shift-F6)

Selects a database from the current list of named ranges.

Select Criteria (Shift-F7)

Selects a criteria from the current list of named ranges.

Sort Database (Shift-F8)

Sorts the current database in increasing order, based on the first field (column).

Sort Column (Shift-F9)

Sorts the current column in increasing order.

Macros Menu

Open Macros (Alt-F1)

Loads a macro set from disk. The new macros are appended to those already in memory.

Save Macros (Alt-F2)

Saves a copy of all macros currently in memory, to disk.

Close Macro (Alt-F3)

Removes a specified macro from memory.

Close ALL Macros (Alt-F4)

Removes all macros from memory.

Start Recording... (Alt-F5)

Creates a new macro. Once this command has been selected, all keystrokes and menu selections are recorded to the macro until *Macros/End Recording* is selected.

End Recording (Alt-F6)

Terminates a macro sequence and prompts you for a name for the macro. Enter a unique name of up to 18 characters and press RETURN. Press RETURN without entering a name to abort the macro creation.

Play Back! (Alt-F7)

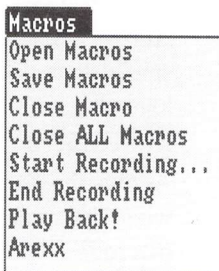
Executes a specified macro. Select the desired macro from the *List* requester and then click on the *Continue* button.

Arexx/Attach

Allows you to attach Arexx programs to individual cells in a spreadsheet. Subsequently, double-clicking on the cell will run the ARexx script, if the ARexx language is available.

Arexx/Detach

Detaches an ARexx script from a cell. If no script was attached to the currently selected cell, this command has no affect.



Arexx/Execute

Used for testing Arexx scripts without attaching them to cells in your spreadsheet.

Auto/Macro File

Allows a macro file to be "attached" to a spreadsheet. After a spreadsheet is loaded, then this macro file will be automatically loaded.

Auto/Macro

Allows the user to specify that a macro be automatically executed when the spreadsheet is loaded. The user should make sure that the macro file chosen is one from the macro file specified in *Auto/Macro File*.

Auto/ARexx

Allows the user to specify an ARexx script to be played when the spreadsheet is loaded.

The following menus are available from a Chart window.

File		
Save to IFF		I
Save to CAD		C
Save to Draw		D
Print		P
Enhanced Print		E
Close		Y

File Menu

Save to IFF (RightAmiga-I)

Saves the current chart to disk in IFF format. Make sure that the desired chart window is active (click anywhere within the window bounds) before selecting this command.

Save to CAD (RightAmiga-C)

Saves the current chart to disk in *Aegis Draw Plus* format. This format is read by Gold Disk's *Professional Page*, as a structured drawing.

Save to Draw (RightAmiga-D)

Saves the current chart to disk in Gold Disk's *Professional Draw* format. The resulting file, (called a *Clip* in *Professional Draw* terms), can be read by either *Professional Draw* or *Professional Page*.

Print (RightAmiga-P)

Prints the current chart. Use the Preferences tool on your Workbench disk to set up the optional printout characteristics (black and white/gray scale, dithering, etc.) before selecting this command.

Enhanced Print (RightAmiga-E)

Prints the current chart in an enhanced format that shows more detail

than a standard printout. This mode of printing requires more memory than normal printing, but yields better results.

Close (RightAmiga-Y)

Closes the current *Chart* window. A chart window may also be closed by clicking on its *Window Close* gadget.

Chart Control Menu

Change Chart Type (RightAmiga-T)

Opens the *Chart Selection* requester and allows you to choose a new type of chart for displaying the current chart data.

Prefs (RightAmiga-R)

Opens the *Chart Format* requester and allows you to modify the format of the current chart. The available format options are dependent on the chart type.

Set Crossing (RightAmiga-S)

Lets you specify the point (value) at which the chart x-axis "crosses" the y-axis for column, area and bar charts. Although the x-axis is not actually displayed at this point, all bars and areas are drawn outward from this value. The "floating" option, found in the *Chart Format* requester, must be on for this command to have any effect. By default, the x-axis "crosses" at the minimum y value.

Use/Patterns

Uses patterned lines and area fills to build the chart.

Use/Solid Colors

Uses solid lines and areas to build the chart.

Set Max and Min (RightAmiga-M)

Sets the range of values to be plotted on the y-axis.

Data/By Column

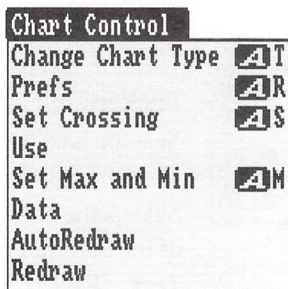
Group data into sets by columns in the selected range of cells.

Data/By Row

Group data into sets by rows in the selected range of cells.

AutoRedraw/On

Update the chart window each time a change is made to the relevant chart data.



AutoRedraw/Off

Disable automatic updating of the chart window. When *AutoRedraw* is Off, you may use *Chart Control/Redraw* to update the chart as desired.

Redraw

Refreshes the *Current Chart* window and updates it to reflect the data currently stored in the chart data range (as defined when the chart was created).

BackGround	 B
Title	 L
Text	 X
Axis	 A

Colors Menu

Background (RightAmiga-B)

Sets the background of the current chart to a selected color.

Title (RightAmiga-L)

Selects the color to be used for drawing the title and axes labels of the current chart.

Text (RightAmiga-X)

Selects the color to be used for drawing the legend and any values on the current chart.

Axis (RightAmiga-A)

Selects the color to be used for drawing the axes of the current chart.

Chart Options

Chart options are used to fine tune the appearance of a chart. These options are modified from the *Chart Options* requester which can be activated by selecting *Chart Control/Prefs* from the menu. The available options will vary based upon the type of chart currently displayed. The following list of options is given in alphabetical order.

All Exploding

Separates the individual slices of a pie graph.

Boxes

Draws a box at the location of each point plotted on the chart.

Drop Lines

Draws lines from each point plotted on the chart, down to the horizontal axis.

First Explosion

Separates the first slice of a pie chart from the remainder of the slices.

Floating

Allows bars to be plotted outward from a point above the axis (for example, to indicate positive and negative values more clearly). Refer to the "Set Crossing" command in Chapter 8, Menu Reference.

Horizontal Axis

Draws the horizontal (x) axis.

Horizontal Grid

Draws horizontal lines across the chart at the position of each tick mark along the vertical axis.

Legends

Displays a legend in the upper right corner of the chart window. The chart itself will be drawn smaller in order to accommodate the addition of the legend.

Overlap

Overlaps the set of bars at each x-position in a bar chart.

Right Vert Axis

Draws a vertical (y) axis at the right side of the chart.

Stacked

Stacks the bars/areas at each position along the x-axis.

Ticks

SKIP is used to indicate the number of tick marks to display along the horizontal axis. Every "SKIP+1"th tick mark will be drawn. For example, if SKIP is set to zero (0), all tick marks will be displayed. If SKIP is set to one (1), every second tick mark will be drawn and so on.

START is used to select the column/row of the spreadsheet which will provide the labels for the horizontal axis of the chart. Set START to the number of the column/row from which you wish to extract the labels. For example, if you set START to one (1), the labels will be extracted from column A (or row 1) of the spreadsheet.

Top Horiz Axis

Draws a horizontal (x) axis at the top of the chart.

Values

Displays the numeric value for each point plotted on the chart.

Vertical Axis

Draws the vertical (y) axis.

Vertical Grid

Draws vertical lines on the chart at the position of each tick mark along the horizontal axis.

BUILT-IN FUNCTIONS

In the function definitions which follow, all references to the function arguments are shown in *italics*.

Definitions

Numeric Expression:

- any formula which reduces to a numeric value
- numeric expressions may make use of the four basic mathematical operators (addition (+), subtraction (-), multiplication (*), and division (/)) as well as the built-in math functions provided by *Advantage*. example: `SQRT( 4.5 * AVG(B1:B3))`
- numeric expressions may be used where boolean expressions are expected. If the expression reduces to a value of zero (0), it is considered FALSE. Otherwise, it is considered TRUE.

Boolean Expression:

A simple boolean expression makes a comparison between two values, either calculated or constant. The result of the comparison is either TRUE or FALSE. Boolean expressions make use of logical operators to perform their comparisons. Examples of the available logical

operators are:

a = b TRUE if a is equal to b

a > b TRUE if a is greater than b

a < b TRUE if a is less than b

a >= b TRUE if a is greater than or equal to b

a <> b TRUE if a is not equal to b

Complex boolean expressions can be built up from simple expressions through the use of the boolean functions (described later in this appendix). For example:

AND(A1 > 5, A1 < 10)

This expression is TRUE if the value in cell A1 is between 5 and 10.

Boolean expressions may also be used where a numeric expression is expected. When used in this manner, a boolean expression has a value of one (1) when it is TRUE and a value of zero (0) when FALSE.

Value:

- a numeric constant example: 5.6
- a cell name referring to a cell containing either a constant or a calculated numeric value
- any expression (numeric or boolean) which reduces to a numeric value

Range:

- a rectangular region within the spreadsheet work area, the boundaries of which are indicated by the names of the cells in the upper-left and lower-right corners of the region separated by a colon. For example, the range A1:C3 includes these cells: A1, A2, A3, B1, B2, B3, C1, C2, C3.

List:

- a series of values separated by commas
example: (1,5,9,7.2*A1)
- a series of cell names separated by commas
example: (A1,B2,C7)
- a range or series of ranges separated by commas

example: (A1:B2) or (A1:B2,C2:D7)

- any combination of the above

example: (A1,5,C2:D7)

Database:

- a range, the contents of which are organized in database format

Criteria:

- a range, the contents of which indicate selection criteria to be applied against records in a database

Database Functions

DAVERAGE(Offset, Database, Criteria)

Returns the average value of the field *Offset* columns from the first column of the *Database* range for all records in *Database* which match the specified *Criteria*. *Offset* must refer to a numeric field in the database records. An *Offset* value of zero (0) will return the average for entries in the first column of the database.

DCOUNT(Offset, Database, Criteria)

Returns the number of records in *Database* which match the *Criteria*.

DDB(Cost, Salvage, Life, Period)

Calculates the depreciation allowance of asset for a specified period.

Uses the double-declining balance method.

DMAX(Offset, Database, Criteria)

Returns the largest value, from the field *Offset* columns from the first column of the *Database* range, of all records in the *Database* which match the *Criteria*. *Offset* must refer to a numeric field.

DMIN(Offset, Database, Criteria)

Returns the smallest value, from the field *Offset* columns from the first column of the *Database* range, of all records which match the *Criteria*. *Offset* must refer to a numeric field.

DSTDEV(Offset, Database, Criteria)

Returns the standard deviation of values in the field *Offset* columns from the first column of the *Database* range, for all records in

Database which match the specified *Criteria*. *Offset* must refer to a numeric field.

DSUM(Offset, Database, Criteria)

Returns the sum of values in the field *Offset* columns from the first column of the *Database* range, for all records in *Database* which match the specified *Criteria*. *Offset* must refer to a numeric field.

DVAR(Offset, Database, Criteria)

Returns the variance of the values in the field *Offset* columns from the first column of the *Database* range, for all records in *Database* which match the specified *Criteria*. *Offset* must refer to a numeric field.

Statistical Functions

All statistical functions ignore blank cells and cells containing text in computing their values. This allows the calculations to be performed correctly even if one or more such cells are embedded within the range of cells included in the *List* argument. .

AVG(List)

Returns the average of all numeric values in *List*.

COUNT(List)

Returns the number of items in *List* which have a numeric value.

MAX(List)

Returns the largest value in *List*.

MIN(List)

Returns the smallest value in *List*.

STDEV(List)

Returns the standard deviation of all numeric values in *List*.

VAR(List)

Returns the variance of all numeric values in *List*.

Financial Functions

When using the financial functions keep in mind that all values should be specified for the same time interval. For example, if the payment amount is given as a monthly figure, then the interest rate should also

be given on a monthly basis and the number of periods should be specified in months. For calculations where the payment amounts are indicated by a Range, the cells in the range are assumed to be equally spaced in time. You can, of course, create an unequally spaced series of payments by entering zeroes as required.

CTERM(Interest, Future Value, Present Value)

Calculates the number of periods it takes for an investment (present value) to grow to a future value earning a fixed interest, and compounding at the end of each period.

FV(Rate, Number of Periods, Payment)

Returns the future value of a series of equal payments. Rate indicates the periodic interest rate, Number of Periods indicates the number of payments to be made and Payment indicates the amount of each payment.

FVV(Rate, Range)

Returns the future value of a series of payments. The payment amounts are usually (though not necessarily) unequal in this case, since FV() may be used for a series of equal payments. *Rate* indicates the periodic interest rate. *Range* indicates a series of cells containing the periodic payment amounts.

IRR(Guess, Range)

Returns the internal rate of return (IRR) for an investment. *Guess* indicates an initial guess for the IRR calculation. *Range* indicates a series of cells containing both the outflows (negative values) and inflows (positive values) of money from the investment. There must be at least one negative and one positive amount in *Range* for the calculation to be successful.

LRATE(Future Value, Present Value, Term)

Calculates the interest rate required for an investment (Present Value) to grow to a Future Value, over Term compounding periods.

NPER(Rate, Principal, Payment)

Returns the number of payments ("number of periods") necessary to pay off a loan of size *Principal* at a periodic interest rate of *Rate* and a periodic payment amount of *Payment* dollars. If *Payment* is too small (not enough to cover the interest due and thus will never pay off the principal) an error message is returned.

NPV(Rate, Range)

Returns the net present value (NPV) of a series of cash flows. Unlike the present value function (*PV()*), the net present value allows for both outflows (negative values) and inflows (positive values) of cash. *Rate* indicates the periodic interest rate. *Range* indicates a series of cells containing the cash inflows and outflows for which the NPV is to be determined.

PMT(Rate, Principal, Number of Periods)

Returns the periodic payment amount required to repay a loan of size *Principal* at periodic interest rate *Rate* over *Number of Periods* payments. Payments are to be made at the beginning of each period.

PV(Rate, Number of Periods, Payment)

Returns the present value of *Number of Periods* payments, each of size *Payment* with a periodic interest rate of *Rate*.

RATE(Guess, Payment, Number of Periods, Present Value, Future Value)

Returns the rate of return for an investment that generates a series of *Number of Periods* payments, each of size *Payment*. *Guess* indicates an initial guess at the rate of return.

SLN(Cost, Salvage, Life)

Calculates the straight-line depreciation allowance of an asset for one period.

SYD(Cost, Salvage, Life, Period)

Calculates the sum of the years' digits depreciation allowance of an asset for a specified Period.

TERM(Payments, Interest, Future Value)

Calculates the number of payments to accumulate Future Value. Assumes

one Payment per period, payments of equal value, and compounding interest.

Boolean Functions

Boolean functions allow you to perform "decision making" in your spreadsheet calculations. These functions treat values as either TRUE (non-zero) or FALSE (zero). In conjunction with the IF function, these

functions allow you to present different results based on the status of various cells on your spreadsheet.

AND(List)

Returns TRUE if all items in *List* are TRUE. Otherwise, returns FALSE.

FALSE()

Returns 0.

IF(Boolean Expression, True Expression, False Expression)

If the *Boolean Expression* reduces to TRUE, returns the value of *True Expression*. Otherwise, returns the value of *False Expression*.

ISER(Cell)

Returns TRUE if *Cell* has an error. Otherwise, returns FALSE.

ISNA()

Same as ISERR(). Added for 123 compatibility.

ISNUMBER(Cell)

Checks whether *Cell* contains a valid number. If so, returns 1, else it returns 0.

ISNV(Cell)

Returns TRUE if *Cell* does NOT contain a value. Otherwise, returns FALSE.

ISSTRING(Cell)

Checks whether *Cell* contains a string. If so, returns 1, else it returns 0. IF ISSTRING references formula cell that returns a string then it will return 1.

NA()

Same as ERR(). Added for 123 compatibility.

NOT(Value)

Returns TRUE if *Value* is FALSE. Returns FALSE if *Value* is TRUE.

OR(List)

Returns TRUE if any item in *List* is TRUE. Otherwise, returns FALSE.

PRINTIF(Boolean Expression, True String, False String)

If the *Boolean Expression* reduces to TRUE, displays *True String* in

the cell. Otherwise, displays *False String*.

SAYIF(Boolean Expression, True String, False String)

If the *Boolean Expression* reduces to TRUE, displays *True String* in the cell and reads the text out loud. Otherwise, displays/says *False String*.

TRUE()

Returns 1.

XOR(List)

Returns TRUE if there is a mixture of TRUE and FALSE items in *List*. If items in *List* are either all TRUE or all FALSE, returns FALSE.

General Math Functions

ABS(Value)

Returns the absolute value of *Value*.

E()

Returns the value of the constant 'e' (2.7182818285).

EXP(Value)

Returns e raised to the power *Value*.

FAC(Value)

Returns the factorial of *Value*.

FIB(Value)

Returns the fibonacci number of *Value*.

INTEGER(Value)

Truncates the fractional part of *Value* and returns only the integer portion.

LN(Value)

Returns the natural log (log base e) of *Value*.

LOG(Value)

Returns the log base 10 of *Value*.

LOGA(Base, Value)

Returns the log base *Base* of *Value*.

MOD(Value, Divisor)

Returns the remainder of *Value* divided by *Divisor*.

POWER(Value, Power)

Returns *Value* raised to the *Power* power.

RAND()

Returns a pseudo random number between 0 and 1.

ROUND(Value, Digit Position)

Returns *Value* rounded to *Digit Position*. *Digit Position* indicates a "column number" relative to the decimal point, where rounding should be done. For example, -1 will round to one decimal place, 0 will round to no decimal places and 1 will round to one digit to the left of the decimal place (ie. to the nearest 10).

SIGN(Value)

Returns 1 if *Value* > 0, -1 if *Value* < 0 or 0 if *Value* = 0.

SQRT(Value)

Returns the square root of *Value*.

SUM(List)

Returns the total of all values in *List*.

Trigonometric Functions

ACOS(Value)

Returns the arc cosine of *Value*.

ACOSH(Value)

Returns the arc hyperbolic cosine of *Value*.

ASIN(Value)

Returns the arc sine of *Value*.

ASINH(Value)

Returns the arc hyperbolic sine of *Value*.

ATAN(Value)

Returns the arc tangent of *Value*.

ATANH(Value)

Returns the arc hyperbolic tangent of *Value*.

COS(Value)

Returns the cosine of *Value*.

COSH(Value)

Returns the hyperbolic cosine of *Value*.

DEGTORAD(Value)

Converts *Value* from degrees to radians.

PI()

Returns the value of the constant 'pi' (3.1415926536)

RADTODEG(Value)

Converts *Value* from radians to degrees.

SIN(Value)

Returns the sine of *Value*.

SINH(Value)

Returns the hyperbolic sine of *Value*.

TAN(Value)

Returns the tangent of *Value*.

TANH(Value)

Returns the hyperbolic tangent of *Value*.

Index Functions

CELL(Row Offset, Column Offset)

Returns the contents of the cell *Row Offset* rows and *Column Offset* columns from the cell where this function is used.

CHOOSE(Offset, Range)

Returns the item *Offset* entries into *Range*. An *Offset* of one returns the first entry in *Range*. If *Range* contains more columns than it does rows, then the entries are ordered by rows. That is, *Offset* 1 returns the first entry in the first row, *Offset* 2 returns the second entry in the first row, etc. To return the first entry in the second row, *Offset* must be equal to the length of a row plus one. Similarly, if *Range* contains more rows than columns, the entries are ordered by columns, so that the first entry of the second column follows the last entry of the first column. If the number of rows and columns are equal, the entries are ordered by columns.

COL(Value)

Returns the column number of the cell plus *Value*.

HLOOK(Offset, Lookup Range, Search Value)

Searches through the first row of *Lookup Range*, from left to right, for the first entry greater than *Search Value*. Call the column where this entry occurs, the Stop Column. HLOOK returns the value *Offset* rows into *Lookup Range*, from the column to the **left** of the Stop Column. If the cell is empty, a "No value in cell" error will be returned.

INDEX(Range, Row Offset, Column Offset)

Returns the contents of the cell which is offset from the upper-left corner of *Range* by *Row Offset* rows and *Column Offset* columns.

LCELL(Name, Cell)

Returns the value of cell *Cell* in spreadsheet *Name*. This allows you to link calculations between spreadsheets. Spreadsheet *Name* must be loaded into memory for this function to operate. Note that *Name* is simply the file name portion of the spreadsheet, not the entire path name (ie. use "Tutorial.ADV" rather than

"**Advantage**:Examples/Tutorial.ADV").

LRANGE(Name, Range)

Returns the range of cells *Range* from spreadsheet *Name*. This allows you to link calculations between spreadsheets. Spreadsheet *Name* must be loaded into memory for this function to operate. Use only the file name of the spreadsheet as *Name*, not the entire path name (see LCELL, above).

N(Range)

Returns the entry in the first cell of *Range* as a value.

RANGE (Row1, Column1, Row2, Column2)

Returns the range of cells delimited by the cells *Column1*
Row1:Column2 *Row2*.

RIGHT(String, Value)

Returns the rightmost *Value* characters.

ROW(Value)

Returns the row number of the cell plus *Value*.

S(Range)

Returns the first cell in *Range* as a string.

VLOOK(Offset, Lookup Range, Search Value)

Searches through the first column of *Lookup Range*, from the top down, for the first entry greater than *Search Value*. Call the row where

this entry occurs, the Stop Row. VLOOK returns the value *Offset* columns into *Lookup Range*, from the row **above** the Stop Row. If the cell is empty, a "No value in cell" error will be returned.

Time Functions

DATE(Month, Day, Year)

Returns the number of days between January 1, 1978 and the date indicated by month number *Month* (1 to 12), day number *Day* (1 to 31) and year number *Year*. Use the menu command *Format/Date* to display this value in a text format.

DATEVALUE(String)

Converts the string to internal **Advantage** Date format

HOUR()

Returns the number of hours since midnight to the time when this cell was last recalculated.

LHOUR(Value)

If Value is in **Advantage** internal time format, returns the Hour of Value.

LMINUTE(Value)

If Value is in **Advantage** internal time format, returns the Minute of Value.

LSECOND(Value)

If Value is in **Advantage** internal time format, returns the Second of Value.

MINUTES()

Returns the number of minutes since midnight to the time when this cell was last recalculated.

MONTHDAY(Value)

Returns the day of the month associated with *Value*, where *Value* is the number of days since January 1, 1978.

MONTH(Value)

Returns the month number associated with *Value*, where *Value* is the number of days since January 1, 1978.

NOW()

Returns the time of day (as a fraction of a day) when the cell contents were last recalculated. Use *Format/Time* to display this value as a time of day in AM/PM format.

SECONDS()

Returns the number of seconds since midnight to the time when this cell was last recalculated.

TIME(Hours, Minutes, Seconds)

Returns the number of "ticks" between midnight and the time of day indicated by *Hours*, *Minutes* and *Seconds*.

TIMEVALUE(String)

Converts String into internal time format.

TODAY()

Returns the number of days since January 1, 1978 to the time when this cell was last recalculated. Use *Format/Date* to display this value in a text format.

WEEKDAY(Value)

Returns the day-of-the-week number associated with *Value*, where *Value* is the number of days since January 1, 1978. (Sunday = 0, Monday = 1, etc.)

YEAR(Value)

Returns the year number associated with *Value*, where *Value* is the number of days since January 1, 1978.

Miscellaneous Functions

CHAR(Value)

Returns the ascii character equivalent of *Value*. Accepts values between 32 and 255.

CLEAN(String)

Removes control characters from strings. Used to "clean" text imported from word processors that have formatting characters.

CODE(String)

Returns the ascii value of the first character in the string.

COLOR(Boolean Expression, True Color, False Color)

Returns the value *True Color* if Boolean Expression evaluates to TRUE. Otherwise, returns the value *False Color*. In addition, this function changes the text color for the cell in which the function appears, to that of the pen number indicated by the value returned.

ERR()

Sets a cell's error flag.

EXACT(String1, String2)

IF both strings are exactly the same, returns 1, else returns 0. Use this function when the case of the string is important.

Example:

= "String 1" = "string 1"

this formula would return 1

however,

= EXACT("String 1", "string 1")

would return 0.

FIND(Search String, String, Start Position)

Returns the position in String where Search String is first found. It begins searching at Start Position (0 is the first character).

LCOLS(Range)

Returns the number of columns in the range.

LEFT(String, Value)

Returns first Value characters of String.

LENGTH(String)

Counts the number of characters in String.

LINKDISK(String, Cell)

Returns the value (or string) in the specified cell from a spreadsheet on disk. String is the full pathname of a spreadsheet file on disk. Cell is the cell the user wants the value from.

LOWER(String)

Forces all characters in String to lower case.

LROWS(Range)

Counts the number of rows in RANGE

MID(String, Start Number, Value)

Returns Value characters from String, beginning with the character at Start Number

PROPER(String)

Converts the letters in string to proper capitalization. The first letter in each word is changed to upper case, the rest to lower.

Example:

=PROPER("abc-def TEST")

would return: "Abc-Def Test"

REPEAT(String, Value)

Returns String Value times.

Example:

Repeat("=",5)

would return "=====

REPLACE(Original String, Start, Num Del, New String)

Insert New String in Original String at place Start, after deleting Num Del characters from Original String from place Start.

SETCOLOR(Boolean Expression, True Color, False Color, Cell)

Sets the pen color of cell *Cell* to either *True Color* or *False Color*, based on the result of *Boolean Expression*.

SETSTYLE(Boolean Expression, True Style, False Style, Cell)

Sets the text style cell *Cell* to either *True Style* or *False Style*, based on the result of *Boolean Expression*. Refer to **STYLE()** for a description of the text style codes

STRING(Value, N)

Converts Value into a string with N decimal places..

STYLE(Boolean Expression, True Style, False Style)

Returns the value *True Style* if Boolean Expression evaluates to TRUE. Otherwise, returns the value *False Style*. In addition, this function changes the text style for the cell in which the function appears, to the style indicated by the value returned. The text style values are expressed as binary numbers (one and zeroes only) with specific attributes associated with the digit positions in the number. The right-most bit activates underlining, the middle bit controls bolding and the

leftmost bit controls italics. For example:

0 = plain text

1 = underline

10 = bold

100 = italics

110 = bold and italics

TRIM(String)

Removes all leading, trailing and consecutive spaces from String.

TYPE(Cell)

Returns a value indicating the type of information stored in cell *Cell*.

- 0: Error or cell is not there
- 1: Cell contains a formula
- 2: Cell contains a selection criteria formula
- 3: Cell contains a value
- 4: Cell contains a label

UPPER(String)

Converts all characters in String to upper case.

VALUE(String)

Converts String into a number.